

Fast, compact routine interfaces EEPROM to μ C

Grzegorz Mazur, Institute of Computer Science, Nowowiejska, Poland

THE CODE in Listing 1 provides the interface between any MCS-51 family of μ Cs and a 24C01a/2/4/8/16 I²C EEPROM. The interface is purely software-driven, so any μ C port pins can control the EEPROM. This code is approximately two times smaller and approximately 20% faster than similar routines published by Atmel Corp (www.atmel.com). Careful coding of low-level routines and structural optimization pro-

duce the performance improvements.

The code is tuned for a -51 running at 12 MHz, but you can also use it at lower clock speeds. For higher speeds, you must adjust low-level routines by inserting "nop" instructions to match I²C-specified timings. The maximum data rate while reading memory content as 16-byte blocks is as high as 8.1 kbytes/sec. With minor modifications, you can use the routines to interface any I²C slave device

to a -51 μ C.

Listing 1 is available for downloading from EDN's Web site, www.ednmag.com. At the registered-user area, go into the "Software Center" to download the file from DI-SIG, #2329. (DI #2329).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 398

LISTING 1—I²C EEPROM- μ C-INTERFACE ROUTINE

```

DEVADDR equ 0a0h ; EPROM I2C device address

setl bit p3.4
sda bit p3.5

; register usage
; r0 buf_ptr ; pointer to internal RAM buffer for block transfers
; r1 count ; byte count for block transfers
; r2 addr_lo ; 2-byte memory address (up to 11 bits used)
; r3 addr_hi
; r4 scratch ; used internally

write_block: ; write block from RAM to EEPROM
    acall send_full_address
    jc fin
wblock:
    mov a, #0
    inc r0
    acall shift_out
    jc stop
    djnz r1, wblock
    sjmp stop

write_byte: ; write single byte from ACC to EEPROM
    push acc
    acall send_full_address
    pop acc
    jc fin
    acall shift_out
    sjmp stop

read_block: ; read block from EEPROM to RAM
    ; 342 + 102 * nbytes
    acall send_full_address
    jc fin
    acall send_devaddr_rd
    jnc rbl
    ret

rblock:
    mov a, #0ffh ; input data from SDA and send ACK
    acall shift
    setb sda
    mov #0, a
    inc r0
rd1: djnz r1, rblock ; 2 + 102 * nbytes
    acall shift_in_nak
    mov #0, a
    inc r0
    sjmp end_read

read_random: ; read random byte from EEPROM to ACC
    ; 448 C
    acall send_full_address
    jc fin

read_currnt: ; read next byte from active EEPROM page to ACC
    ; 224 C
    acall send_devaddr_rd
    jc fin
    acall shift_in_nak

end_read:
    cpl c ; 10 C
stop:
    ; Send STOP
    ; 9 C
    cpl sda
    setb scl

nop
nop
nop
setb sda ; 4.7us after SCL
; SCL & SDA high.
ret

send_full_address: ; 220 C
    cpl c
    acall send_devaddr ; send device address
    jc fin ; send byte address
    mov a, r2
    sjmp soasc

send_devaddr_rd: ; 116 C
    setb c
send_devaddr: ; 111 C
    ; Send START
    setb scl

    jnb scl, bbsy ; Check if bus available
    jnb sda, bbsy
    cpl sda ; min. 4.7 us after SCL high
    mov a, r3
    rlc a
    ori a, #DEVADDR
    cpl scl ; min. 4.0 us after SDA low
    ; SCL & SDA low.
    ; send device id, 3 msbits of address and rw bit (from c)
soasc:
    acall shift_out
    jc stop
fin:
    ret

bbsy:
    setb c ; bus not available
    ret

; Shift out/in 3 byte from/to A, msb first and ACK bit from/to C.
; SCL low, SDA high on entry and exit.
shift_in_nak: ; 96 C
    mov a, #0ffh ; fill ones - input data from SDA
    shift_out: ; 95 C
        setb c ; 3rd bit - output NAK/release SDA to
        shift: ; 94 C
            mov r4, #9 ; 9 bits + ack
        shlop: ; 3.5 us CLK low to data valid
            rlc a ; move bit into C
            mov sda, c
            setb scl
            nop
            nop
            mov c, sda
            cpl scl ; min. 4 us
            djnz r4, shlop
            ret ; ACK in C
end

```

Photodetector sorts objects

Alan Erzinger, Harris Semiconductor, Palm Bay, FL

MOST OBJECT-SENSING SYSTEMS have problems detecting the presence of an object. The system in **Figure 1** uses an oscillator to ease detection problems and allow sorting. The oscillator reduces power dissipation in the photodiode by operating the diode at 50% duty cycle. The oscillator also enables a 50% increase in the noise-filter time constant, and it functions as a time-base to allow object sorting. The oscillator chops the photodiode's bias. The signal the photodetector receives is a square wave; thus, a filter can remove optical noise. You normally need filtering when light shines through fans onto the optical detector. When required, you can place a bandpass filter in series with Q_3 's output.

The oscillator frequency has two limits: the response time of the phototransistor and the accuracy of the object-sensing system. Q_4 and Q_3 are connected in a cascode configuration to minimize the Miller effect in Q_4 . This connection reduces the pair's optical transient response to nanoseconds, thus allowing os-

cillator periods of submicroseconds. Objects on a conveyor belt travel at relatively low speeds. You can calculate their expected time in front of the photodetector, t , from $t=d/s$, where s is the conveyor speed in feet per second and d is the object width in feet.

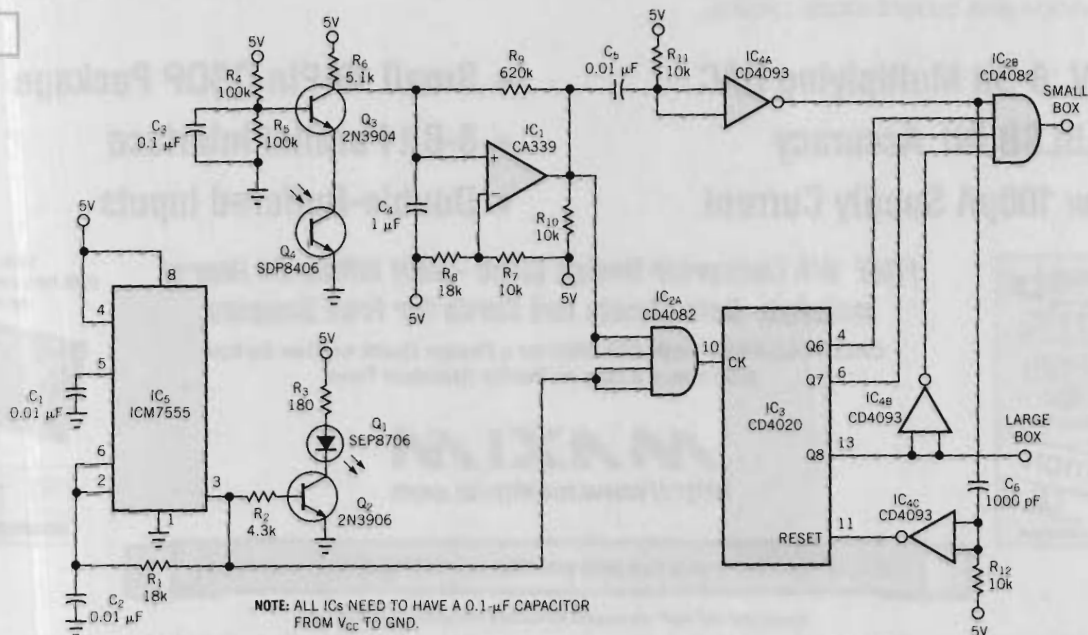
An object 2 in. long with a belt speed of 3.5 ft/sec blocks the detector for 47.6 msec. If the oscillator period is 250 μ sec, the object blocks the detector for approximately 200 periods, so each period equates to 0.5% length accuracy. The system senses two objects—one 2 in. long and one much longer—so 0.5% accuracy is more than adequate. When the detector is unblocked, the inverting input of IC_1 is dominant, and it keeps the output of IC_1 low. The low state prevents the oscillator's output from reaching the counter (IC_3). Blocking the detector allows C_3 to charge to 5V through R_6 , and the noninverting input of IC_1 becomes dominant, starting the count. When the detector is unblocked, the one-shot comprising R_{11} , C_5 , and IC_{4A} pulses IC_{2B} with

an end-of-object pulse. The one-shot's trailing edge triggers the counter-reset one-shot comprising R_{12} , C_6 , and IC_{4C} .

C_4 and R_6 form a nuisance filter that rejects short optical noise. The timing is such to enable a 2-in. object to clear the detector while both Q_6 and Q_7 are high. When the end-of-object pulse coincides with Q_6 high, Q_7 high, and Q_8 low, the output of IC_{2B} indicates a 2-in. object. This situation is the only time window that can indicate a 2-in. object. If the object is longer than 2 in., Q_8 goes high, indicating a large object. When the object clears the detector, the reset one-shot resets the counter for another cycle, and Q_3 quickly discharges C_4 in preparation for another cycle. With the component values shown, the system can sense and discriminate between objects as short as 2 in., separated by 0.1 in. (DI #2325).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 399

Figure 1



An oscillator circuit allows a photodetector to both count and sort objects according to size.

Single-button lock provides high security

Maxwell Strange, Fulton, MD

FIGURE 1 IS THE BLOCK DIAGRAM of an easily programmed, single-button combination lock. You operate the lock by using a series of short and long pulses from a momentary switch that masquerades as a doorbell button. The circuit uses inexpensive CMOS logic. The retriggerable timer, T_2 , locks out entries made after the T_1 code-entry window, thereby greatly enhancing security. The circuit in Figure 2 operates as follows: The Schmitt-trigger quad NAND gate, IC_1 , debounces the code-entry switch and, with the aid of simple analog circuitry, produces separate outputs for activation times of less than and more than 0.3 sec. These outputs connect to the

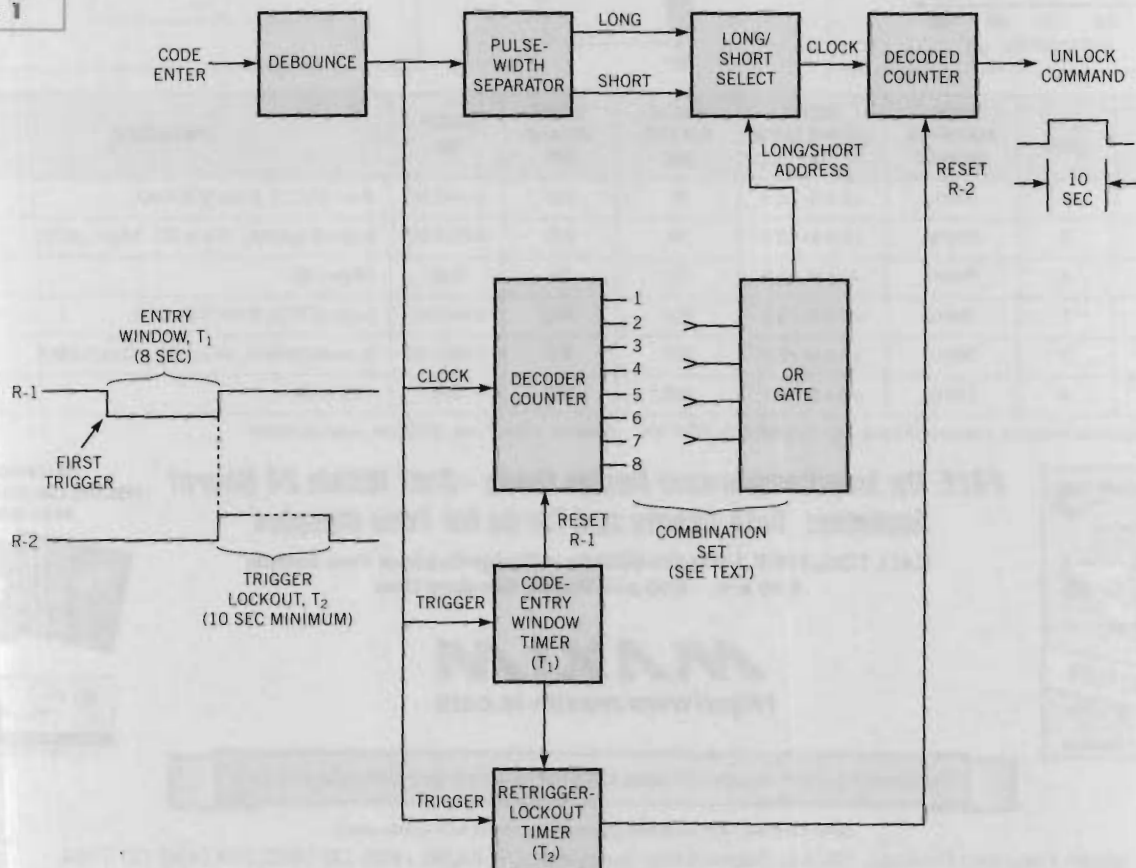
select gate, IC_5 . The initial entry also sets timer T_1 to enable the decoded decade counter, IC_3 . Each entry clocks IC_3 .

As IC_3 steps through its counts, certain of its output positions represent "short" and connect to IC_4 's inputs; unconnected lines represent "long" positions. This coding arrangement sets the combination. Short pulse positions change the address of IC_5 to select the short input pulse; otherwise, IC_5 selects the long pulse input. The short and long inputs, if present in the programmed sequence, produce an output from IC_5 . IC_6 counts the outputs and produces an unlock command only if it counts all pulses. The power-on-reset circuit ensures that no

compromise of security arises under any conditions after a power outage. The timers are crucial to the high security of the system. You must enter the code within the 8-sec T_1 window. If you make a mistake, you must wait at least 10 sec for T_2 to time out before you make another attempt. If entries occur continuously and less than 10 sec apart, as an intruder might try, T_2 continuously inhibits counter IC_6 .

The lock proves to be reliable over several years of use. The circuit in Figure 2 uses an eight-character combination, which you can quickly enter. A short pulse is a quick jab to the button; a long pulse is only slightly longer. A shorter se-

Figure 1

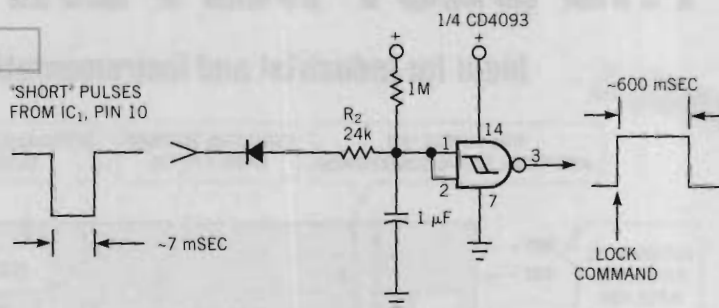


A handful of timers and counters configures a highly secure, single-button combination lock.

quence would also be secure; you can implement a shorter code by simply taking the unlock pulse from a lower count on IC₆. IC₆'s output returns low after 10 sec when T₂ resets. If desired, you can generate a lock command, which need not be secure, by adding the simple circuit in Figure 3. (DI #2327).

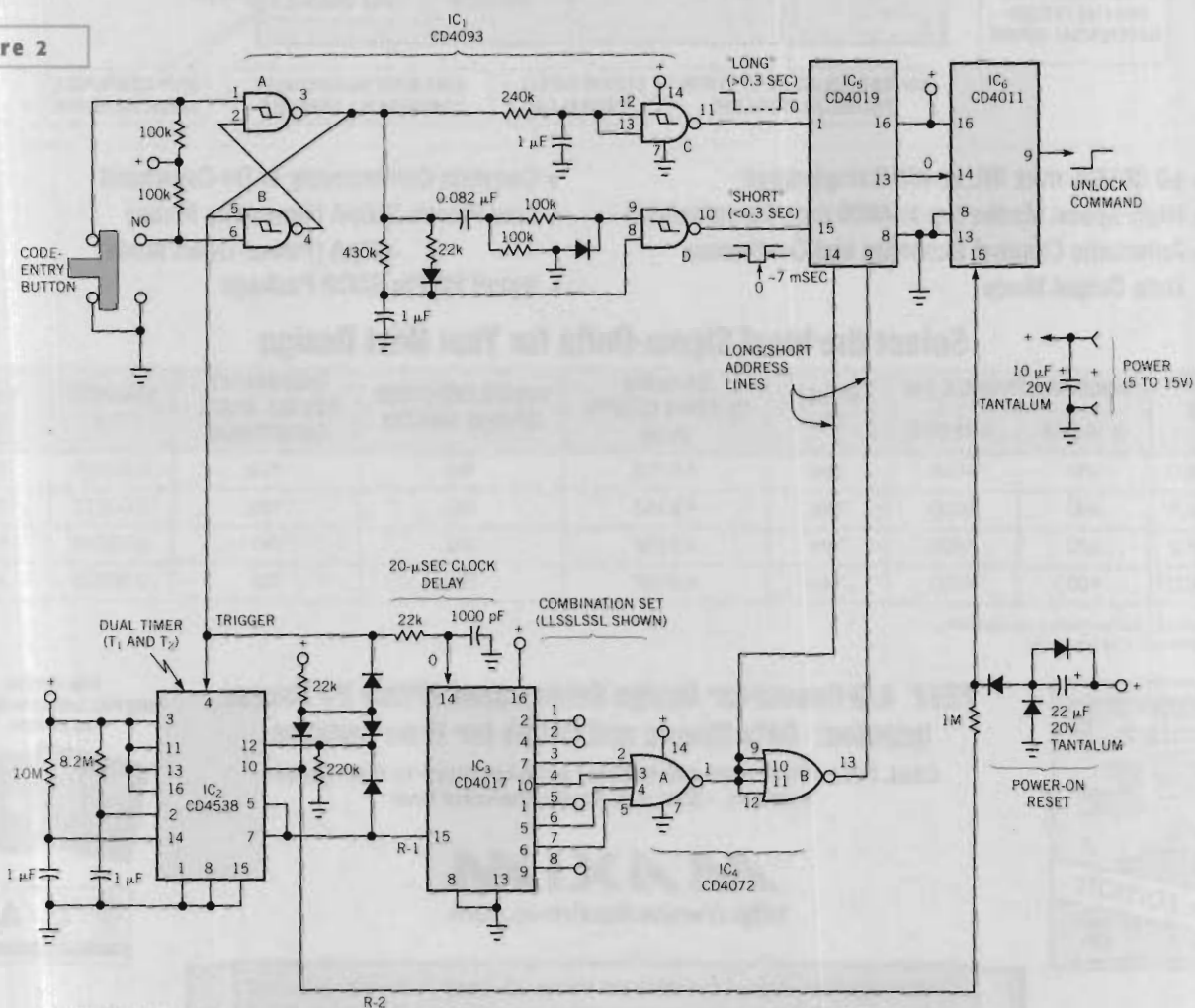
TO VOTE FOR THIS DESIGN,
CIRCLE NO. 400

Figure 3



You can generate a lock command with this additional circuit by rapidly entering four or more short pulses.

Figure 2



NOTE: ALL DIODES=1N4148.

You program your combination by hard-wiring the IC₃-IC₄ output-to-input connections, LLSSLSS, where L and S are long and short inputs, respectively, in this example.